IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	Confirmation No. 1489
Alexandre Kravtchenko et al.	Art Unit: 2194
Serial No.: 09/930,160	Examiner: Diem K. Cao
Filed: August 16, 2001	Customer No. **32658**
For: **IMPORT/EXPORT UTILITY AND A METHOD OF PROCESSING DATA USING THE SAME**	Docket No. P5795

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 CFR § 41.37

I. Real Party in Interest

Sun Microsystems, Inc.
4120 Network Circle
Santa Clara, CA 95054
USA

II. Related Appeals and Interferences

No other appeals or interferences are currently known to Appellants that will directly affect, be directly affected by, or have a bearing on the decision to be rendered by the Board of Patent Appeals and Interferences in the present appeal.

III. Status of Claims

Claims 1, 3, 4, 6, 7, 9-13, 15-18, and 21-34 are pending in the application, with claims 2, 5, 8, 14, 19, 20, and 35 being cancelled. No claims have been allowed, and all pending claims stand rejected under 35 U.S.C. §103. The rejection of claims 1, 3, 4, 6, 7, 9-13, 15-18, and 21-34 is the subject of this appeal.

IV. Status of Amendments

No claim amendments were filed subsequent to the final rejection mailed April 18, 2006, and all claim amendments have been entered.

Claims 1, 3, 4, 6, 7, 9-13, 15-18, and 21-34 are provided in the attached Claims Appendix.

V. Summary of Claimed Subject Matter

Claims 1 and 33 are independent claims that are being appealed.

As noted in para. [0003] of the application, the invention addresses problems associated with prior E-commerce systems in which a database loader was used to load information but only in the form of tables. With reference to para. [0036], the invention uses an import/export utility, that knows nothing of the E-commerce system, to deliver information from an external data file to a business object that does the "real work" such as performing tasks including adding, deleting, or updating the data. The changes in the data can be performed without requiring the business object to be changed. As shown in Fig. 1, the import/export utility communicates with the business object and gathers the imported file and provides a database that is used during the processing of the import file data by the business object. In this manner, the business object is able to import/export data, such as with relation to an E-commerce system, without use of a standard database loader and, hence, to load data not in the form of a table.

With this background in mind, claim 1 is directed to a method of processing data with a utility. Figure 3 of the specification describes such a method of the invention with reference to the systems of Figures 1 and 4 and description provided in paras. [0014] to [0022]. The method includes selecting a file, such as step 22 of Figure 3, that includes a name of a business object. As described in para. [0024], the import file includes a name of the business object to invoke and may also include operations to be performed and properties of the business object.

The method of claim 1 further includes uploading the file to a server and storing file data in a database of the utility (see steps 23 and 24 in Figure 3). The method continues with

delivering the data to the business object and performing a task on the data with the business object (see steps 25 and 26) as this step includes "invoking a code included in the business object, wherein the task comprises adding, deleting, or updating of the data delivered to the business object." As described in para. [0036], the utility may know nothing of an E-commerce or other session it is initiating because the "real work is done inside of the business object itself" with the utility being mainly a "deliverer of information" while the "business object performs the validation and performs the tasks, such as adding, deleting, or updating of the data" that can be done without requiring changing the business object. The method of claim 1 also calls for the code of the business object to include an interface to support an export operation, which is shown in Figure 4 to be provided to the business object by the utility at (5).

Independent claim 33 is directed to a method of loading information to a network application and includes steps of operating a utility to invoke a business object associated with a business object name from an import file. This method is explained generally in para. [0036] and in more detail in the specification with reference to Figure 3 as discussed with reference to the method of claim 1. In the method of claim 33, the business object performs an operation on data delivered by the utility and the method further includes updating the data in a utility database based on the performance of the operation of the invoked business object. Again see para. [0036] and the business object (3), utility (1), import file (4), and export/output file (6) of Figure 4. The operating of the utility to invoke the business object and the updating of the data steps are similar to limitations presented in claim 1, and the summary of the subject matter of claim 1 is applicable to claim 33.

VI. Grounds of Rejection to be Reviewed on Appeal

Claims 1, 3, 4, 6, 7, 9-13, and 21-34 stand rejected under 35 U.S.C. §103(a) as unpatentable over U.S. Pat. No. 6,163,781 ("Wess") in view of U.S. Pat. No. 5,899,998 ("McGauley") further in view of U.S. Pat. No. 6,363,388 ("Sprenger").

VII. Argument

Rejection of Claims 1, 3, 4, 6, 7, 9-13, 15-18, and 21-34 Under 35 U.S.C. §103 is Improper

In the final Office Action of April 18, 2006, claims 1, 3, 4, 6, 7, 9-13, and 21-34 were rejected under 35 U.S.C. §103(a) as being unpatentable over Wess in view of McGauley further in view of Sprenger. This rejection is traversed based on the following remarks, and Appellants request that the rejection be reversed as not properly supported.

As noted in the Summary of the Claimed Invention, the invention is addressing the problems associated with prior E-commerce systems in which a database loader was used to load information but only in the form of tables. With reference to para. [0036], the invention uses an import/export utility, that knows nothing of the E-commerce system, to deliver information from an external data file to a business object that does the "real work" such as performing tasks including adding, deleting, or updating the data (see, for example, claim 35). The changes in the data can be performed without requiring the business object to be changed. As shown in Fig. 1, the import/export utility communicates with the business object and gathers the imported file and provides a database that is used during the processing of the import file data by the business object. In this manner, the business object is able to import/export data, such as with relation to an E-commerce system, without use of a standard database loader and, hence, to load data not in the form of a table.

Turning to specific claim language, claim 1 is directed to a method of processing data with a utility. The method includes selecting a file that includes a name of a business object, uploading the file to a server, storing file data in a database of the utility, delivering the data to the business object, and performing a task on the data with the business object. Also, the method of claim 1 calls for the business object code to provide an interface to support an export operation and for the task performed by the business object to include adding, deleting, or updating of the data delivered to the business object. The combination of Wess, McGauley, and Sprenger fail to teach or suggest all of these elements. Hence, the rejection of claim 1 is improper and should be reversed.

More specifically, the final Office Action cites Wess at col. 6, lines 11-14, col. 6, lines 16-29 and 53-57, and col. 13, lines 3-10 for teaching selecting a file that includes a name of a business object. Appellants disagree with this construction of Wess. Wess at col. 2, lines 30-40 makes it clear that its method is addressing problems associated with having a database with many null data values, and beginning at line 6, col. 4, Wess provides a brief summary of its method that uses a table of defined variable symbols and comparing operations to try to reduce the amount of null data values in its relational databases. Hence, Wess is addressing a different problem than Appellants and uses a different technique to address that problem.

In the cited col. 6, lines 16-29, Wess is describing the converting of data received at a network interface 112 including "textually-based data objects" into formats expected by the system, but at this citation and elsewhere, Wess fails to teach selecting a file that has a business object name in the file. In the claim, the named business object is then used to perform processing on the data of the file, and this is not shown by Wess. Further, Wess fails to suggest that its "data objects" are business objects as defined by Appellants (for example, see paras. [0003] and [0013]). For at least this reason, claim 1 is allowable over Wess.

The Response to Arguments states that "Wess teaches a file that has an object name in the file (see the Fig. 7 and associated text)" and "In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., the data object is not the same as the business object as defined by the Applicant) are not recited in the rejected claim(s)." As to the first statement/response, Appellants' remarks clearly point out that at the cited col. 6, lines 16-29 Wess describes converting data received at a network interface including "textually-based data objects" into formats but fails to teach selecting a file that has a business object name in the file that is then used to perform processing on the data of the file. The Examiner appears to be trying to use the textually-based data objects as both the file that is selected or received and also teaching the business object that is later used to perform a task on that data by invoking code in the business object as called for in claim 1. Figure 7 of Wess shows an example of "five data object instances processable by the system of FIG. 1" (per col. 5, lines 42-43). These appear to be very similar to the file or image file described in Appellants' specification beginning toward the end of page 8,

but fail to show a file that is selected to include a name of a business object that has code to process the data in the file.

In the Amendment of June 13, Appellants asked the Examiner to provide a citation to Figure 7 or elsewhere in Wess where the name of a business object is included in the data object instances of Wess or to withdraw the rejection. In the Advisory Action, the Examiner again cited Fig. 7 and associated text as providing the relevant teaching and stated that in "this case, the name of the business object is 'person identification number.'" Appellants believe that this proves their point that the Examiner is using the data file or object as both the selected file of claim 1 as well as the business object. Figure 7 illustrates a set of data object instances for patients. In col. 7, the object instances are described in detail and no mention of a "person identification number" is provided, but more importantly, there is no discussion of a name of a business object that should be used to perform tasks in the data provided in the data object instances. In other words, looking at the elements 200, 202, 204, 206, and 208, Appellants can find no identification of a business object or anything that could be thought of as a "name" for a business object. The names Smith and Jones are described in col. 7 as providing a link of the other data fields to a particular patient or individual but do not tell a utility a business object to invoke to perform tasks on the data in the data objects of Fig. 7. The identifiers O_1 to O_5 identify the data objects but do not identify another business object (or even another data object) that can be invoked to perform tasks on the data objects but instead identify the objects themselves. Appellants believe that the Examiner has failed to show the selecting step of claim 1 because no file is shown as being selected that includes a name of a business object, and the arguments and further citations provided in the Response to Arguments and Advisory Action fail to explain how the data objects can be used to determine which other object, such as a business object as called for in the claim 1, should be invoked to perform tasks on the data in the objects O_1 to O_5.

As to the second statement in the Response to Arguments regarding an arguing language not present in claim 1, the business object is defined in the claim as being able to perform a task on delivered data "including invoking a code included in the business object, wherein the task comprises adding, deleting, or updating data delivered to the business object and wherein the code includes an interface to support an export operation." The Response statement seems to be

indicating that Appellants were simply arguing "business" is different from "data" but Wess teaches objects. This ignores an important aspect of Appellants' arguments that the textually-based data object of Figure 7 fail to show the business objects defined in claim 1. Appellants can find no showing in Wess' Figure 7 or elsewhere that the data object instances could process data provided in a different file, the data object instances include code that can be invoked, or that the tasks of adding, deleting, or updating would be performed by such invoked code. Hence, Appellants are not arguing limitations that are not present in the claim but instead are asked the Examiner to replace Wess with a reference that shows the type of objects as defined in claim 1 or to withdraw the rejection as unsupported.

The last three Office Actions issued by the Examiner indicated that Wess fails to teach "starting a session, a business object, delivering the data to the business object corresponding to the name uploaded to the server, with the business object, performing a task on the delivered data, the task performing including invoking a code included in the business object."

Due to these deficiencies in Wess, the November 17, 2005 and April 18, 2006 Office Actions cite McGauley for teaching each of these limitations of claim 1 that are not found in Wess (except for an interface to support export generation for which Sprenger is cited). The following discussion stresses how McGauley fails to teach or suggest the data delivering step and the performing of a task on the delivered data with the business object step. Hence, in addition to Wess failing to show the file selecting step as discussed above, McGauley fails to overcome the deficiencies of Wess that have been acknowledged by the Examiner.

McGauley is cited as teaching business objects (at "update objects 240; col. 8, line 60-col. 9, line 21"), delivering the data to the business object corresponding to the name uploaded to the server (at "the data model...tags attached;" col. 11, lines 60-62), performing a task on the delivered data (at "The update type...new record object;" col. 9, lines 46-52), and also where the task performing includes invoking a code in the business object (at "the update object...audit fields 247;" col. 9, lines 14-21). Appellants disagree with this construction of McGauley.

The "updated objects" in McGauley are data files or data objects that are passed throughout a network to allow a patient's medical records to be kept within a portable data

carrier (PDC) or smart card and/or at databases at service providers (point-of-service (POS) stations). A good summary of the McGauley method is provided from col. 3, line 61 to col. 4, line 51. In this description, it can be seen that "data is transmitted via 'update objects'" in "traditional telecommunication channels." The "core of each update object is an element of information or an item of data that has been generated by a medical transaction at a POS station" (such as an X-ray report or a laboratory test result). "Tags" are provided in the update objects and "rule sets" are provided in the system to "guide each update object to its targeted PDC, POS and administrative databases."

As can be seen from these descriptions, the "objects" discussed in McGauley are data objects that are passed throughout a medical data system, and McGauley teaches using tags in the objects and rules at the POS for making sure that the objects are passed to the correct or proper devices (smart cards carried by the patients and databases maintained at POS stations). However, the update objects do not teach the business objects of claim 1, and their teaching does not overcome the various shortcomings of Wess.

Specifically, the update objects and their handling and configuration (including the "tags" and the rule sets spread over a system) do not teach the method of claim 1, which includes selecting a file that includes a name of a business object (i.e., where does McGauley teach this – as it teaches passing the update objects about a system and does not teach retrieving the name of the update object from a selected file) delivering the data to the business object (the update objects are or include the data so data is not delivered to them except for modifications to update the patient records), and performing a task on the data with the business object (again, the update objects are or include the data and any processing of the data is done by an application at the POS or device from which the PDC is access the system).

Further, the method of claim 1 also calls for the business object code to provide an interface to support an export operation, which is not taught by the update objects, and the task performed by the business object include adding, deleting, or updating of the data delivered to the business object, which is not shown by the update object (which carry the data elements but do not perform these tasks). For these reasons, McGauley does not overcome the deficiencies in

Wess, and the combined teachings of these two references does not make the method of claim 1 obvious.

The Response to Arguments in the April 18, 2006 final Office Action appears to be combining record objects, update objects, and processes that operate within the McGauley system in an attempt to find the "business objects" and the steps of the method of claim 1 related to such business objects. This argument is not persuasive for the reasons provided above. Further, the Response to Arguments asserts that McGauley does teach "performing a task on the delivered data, the task performing including invoking a code included in the business object" by teaching "the type of action [that] is needed to carry out" at col. 9, lines 46-52. However, at this citation, McGauley states "the update type identifier indicates the type of processing action that is intended when the update object reaches its destination database(s)." This does not teach using a business object to perform a task by invoking code in the business object itself. Instead, the data of the update object can be added to a record, a record may be deleted, or a record may be deleted after the data reaches its destination database. However, there is no teaching that the code to perform such actions is provided in the update object (or record object for that matter).

The Advisory Action restates the same construction of McGauley without addressing the specific distinguishing arguments provided by Appellants and discussed above. Further, the Response to Arguments points out that McGauley does not teach and is not cited for teaching the file selecting step. As noted above, Wess fails also to teach this step. For these reasons, McGauley fails to provide the teaching for which it is cited, and, therefore, this reference when combined with Wess fail to teach the method of claim 1.

Further, when the teaching of Sprenger is added to that of Wess and McGauley, the method of claim 1 is not taught nor made obvious. Sprenger is not cited for overcoming the deficiencies of Wess and McGauley, and as a result, claim 1 is believed allowable over these three references because of the failings of Wess and McGauley. The Response to Arguments points out that Sprenger is only cited for teaching code that may include an interface to support an export generation and not the performing step. The following discussion is provided to show that Sprenger fails to overcome the deficiencies of Wess and McGauley such that the combined

teaching of these three references is easily seen to fail to support an obviousness rejection of claim 1.

As discussed in a number of Appellants' Amendments, Sprenger in its Summary and elsewhere makes it clear that its method is a data management method that uses agents and minions (see, Sprenger at col. 8, line 60 and on for a discussion of agents and minions) to perform various data management processes, but nowhere in Sprenger is it taught to perform an operation with a business object named in an import file on data uploaded to a database by a utility. At the cited portions of Sprenger, the use of objects is described but there is no teaching of a utility uploading a file and then updating the data based on the performance of the operation by the invoked business object.

For example, Sprenger at col. 5, lines 1-15 teaches that objects can be instantiated from an access layer by accessing the database and later saved to the database. Sprenger does not teach invoking a business object based on an import file and then, using the invoked business object to perform an operation on data in a utility database. For this reason, the pending claims are believed allowable over the combination of Wess and McGauley in further view of Sprenger.

Further, Sprenger at col. 14, lines 59-65 discusses operation of an "EventLogMinion" but this element does not perform validation of data and does not teach that the business object is named in the file providing the data. With this in mind, Sprenger also fails to teach the limitations of claim 1 including delivering the data to the business object and then performing a task on the data with the business object that changes the data.

Specifically, Wess fails to teach the use of business objects, and at col. 14, lines 59-65, Sprenger states a minion "provides a central point of access for all advisory messages that need to be stored persistently in the database. The "events" in the log are descriptions of some event in time, such as the observed failure of a critical component, or the failure of a job stream to complete. The events stored in the log describe unusual conditions that require the attention of the user...." Appellants could find no suggestion in Sprenger of either delivering data from to the business object or that a task that changes the data is performed on the data by the business

object. For these additional reasons, claim 1 is allowable over the combined teaching of Wess, McGauley, and Sprenger.

Claims 3, 4, 6, 7, 9-13, 15-18, 21-32, and 34 depend from claim 1 and are believed allowable as depending from an allowable base claim. Further, claim 34 calls for the business object to perform validation after receiving the data, and this limitation is not shown or suggested by the cited references. In rejecting claim 14, the Examiner has admitted that Wess fails to teach the business object being responsible for validating the data in the selected file but then cited Sprenger at col. 14, lines 59-65 as providing this teaching. In the Office Action preceding the April 18, 2006 final Office Action, however, the Examiner changed the construction of Wess and cited col. 8, lines 46-60 of Wess for teaching the validation with the business object. But, at this citation, Wess discusses use of a comparator comparing value attributes but as noted above this comparator is not provided in a business object to which data is delivered. Therefore, Wess fails to teach the validation with the business object. For these additional reasons, claim 34 is not shown or suggested by Wess, McGauley, and Sprenger.

Independent claim 33 is directed to a method of loading information to a network application and includes steps of operating a utility to invoke a business object associated with a business object name from an import file. In the method of claim 33, the business object performs an operation on data delivered by the utility and the method further includes updating the data in a utility database based on the performance of the operation of the invoked business object. The operating of the utility to invoke the business object and the updating of the data steps are similar to limitations presented in claim 1, and the reasons for allowing claim 1 over the combined teaching of Wess, McGauley, and Sprenger are believed applicable to claim 33.

More specifically, Wess fails to teach receiving a selection of an import file that includes a name of a business object. This named business object is then used to perform processing on the data of the file, and Wess fails to suggest that its "data objects" are business objects as defined by Appellants. The Examiner's Response to Argument in the final Office Action before filing of an RCE confirmed that Wess only teaches data objects (see, fourth paragraph of Response to Arguments). For at least this reason, claim 33 is allowable over Wess.

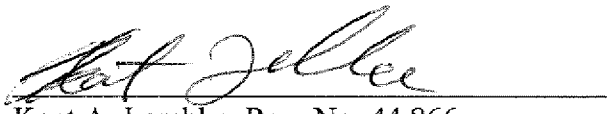
Further, as discussed with reference to claim 1, McGauley does not overcome the deficiencies of Wess, and particularly, provides no teaching of the uploading, operating, and updating steps of claim 33 that involve the business object. Hence, Wess, McGauley, and Sprenger do not support a rejection of independent claim 33.

Conclusion

In view of all of the above, all the pending claims are believed to be allowable and the case in condition for allowance. Appellants respectfully request that the Examiner's rejections based on 35 U.S.C. §103 be reversed for all the pending claims.

Respectfully submitted,

Date: September 11, 2006



Kent A. Lembke, Reg. No. 44,866
HOGAN & HARTSON LLP
One Tabor Center
1200 17th Street, Suite 1500
Denver, Colorado 80202
Phone: (720) 406-5378
Fax: (720) 406-5301

VIII. CLAIMS APPENDIX

1. A method of processing data in a system including an utility, comprising the steps of:

starting a session;

selecting a file on a local drive or by URL, wherein the file includes a name of a business object;

uploading the file including the name of a business object to a server;

storing data of the file in a database of the utility;

delivering the data to the business object corresponding to the name uploaded to the server;

with the business object, performing a task on the delivered data, the task performing including invoking a code included in the business object, wherein the task comprises adding, deleting, or updating of the data delivered to the business object and wherein the code includes an interface to support an export operation; and

downloading and saving a report after the data processing is completed.

3. The method of processing data according to claim 1, wherein the code includes doImport/Export.

4. The method of processing data according to claim 3, wherein the doImport/Export is a command to perform an operation.

6. The method of processing data according to claim 1, wherein the interface includes a code for throwing an attribute set returned by the business object.

7. The method of processing data according to claim 1, wherein the utility is designed to process the file on the remote server.

9. The method of processing data according to claim 1, wherein the session is for import and export operations.

10. The method of processing data according to claim 1, wherein the starting of a session includes generating a unique session ID.

11. The method of processing data according to claim 1, wherein the database of the utility includes a plurality of tables.

12. The method of processing data according to claim 11, wherein the plurality of tables include a user name and session ID storing table, a session status storing table, an initial session data storing table, a result storing table, an error message storing table, and a session log storing table.

13. The method of processing data according to claim 1, wherein the business object includes a code including a command which provides instance of the business object with an attribute.

15. The method of processing data according to claim 1, wherein the business object includes a findByAttributes.

16. The method of processing data according to claim 15, wherein the findByAttributes supports object references in the file.

17. The method of processing data according to claim 1, wherein the business object includes a code that receives a list of validated object attributes and returns a unique object identifier.

18. The method of processing data according to claim 1, wherein the business object includes a code for notifying an end of the data processing.

21. The method of processing data according to claim 1, wherein the file is a text file including command lines, header lines, and data.

22. The method of processing data according to claim 1, wherein the session is an export operation.

23. The method of processing data according to claim 22, wherein the business object exports all of its data or part of the data.

24. The method of processing data according to claim 22, wherein the business object calls an interface including a plurality of codes.

25. The method of processing data according to claim 24, wherein the plurality of codes include a first code including a name of operation and a business object class name to be inserted into an output file.

26. The method of processing data according to claim 24, wherein the plurality of codes include onReceiveExport.

27. The method of processing data according to claim 26, the onReceiveExport is a code for throwing an attribute set returned by the business object.

28. The method of processing data according to claim 1, wherein the utility is a deliverer of information.

29. The method of processing data according to claim 1, wherein the data is changed without changing the business object.

30. The method of processing data according to claim 1, wherein the utility includes a monitoring function, an error handling function, and reporting function.

31. The method according to claim 1, wherein the utility receives information from another system and loads information to the business object.

32. The method according to claim 1, wherein the utility receives information from the business object and stores information on another system.

33. A method of loading information to a network application, comprising:

providing a utility on a server for exporting and importing data to the network application;

associating a database with the utility;

with the utility, receiving a selection of an import file, the import file comprising data in a text file and including a name of a business object;

with the utility, uploading the data and the business object name from the selected import file, wherein the uploaded data is stored in the utility database;

operating the utility to invoke the business object associated with the business object name including delivering at least a portion of the uploaded data to the invoked

business object, whereby the business object performs an operation on the delivered data; and

updating the data in the utility database based on the performance of the operation by the invoked business object.

34. The method processing data according to claim 1, further comprising after the delivering of the data, performing validation with the business object.

IX. EVIDENCE APPENDIX

No copies of evidence are required with this Appeal Brief. Appellants have not relied upon any evidence submitted under 37 C.F.R. §§ 1.130, 1.131, or 1.132.

X. RELATED PROCEEDINGS APPENDIX

There are no copies of decisions rendered by a court or the Board to provide with this Appeal as there are no related proceedings.